Amendments to the Claims

1. (original) A method for providing a graphic representation of code characteristics, the method comprising:

acquiring a block of code in a program;

analyzing the block of code for at least one instruction characteristic;

generating a unique graphical indicator for the at least one instruction characteristic; and

displaying the unique graphical indicator with the block of code to indicate that the at least one instruction characteristic is present in the block of code.

2. (original) The method of claim 1, wherein the at least one code characteristic is selected from the group consisting of a user-visible sub-statement instruction, a loop entry instruction, a loop body instruction, dead code instruction, and a data-speculative load instruction.

3. (original) The method of claim 1, wherein the unique graphical indicator is selected from the group consisting of text color, background color, a line, an arc, a box and a tag.

4. (original) The method of claim 1, wherein the displaying the unique graphical indicator step further comprises:

indicating if the at least one instruction characteristic is a loop-carried dependency.

5. (original) The method of claim 1, wherein the displaying the unique graphical indicator step further comprises:

indicating if the at least one instruction characteristic is a data-speculative load instruction with at least one possible conflicting store.

6. (currently amended) A system for providing a graphic representation of code characteristics tangibly embodied on a computer readable medium, comprising:

means for acquiring a block of code in a program;

means for analyzing the block of code for at least one instruction characteristic;

means for generating a unique graphical indicator for the at least one instruction characteristic; and

means for displaying the unique graphical indicator with the block of code to indicate that the at least one instruction characteristic is present in the block of code.

7. (original) The system of claim 6, wherein the at least one code characteristic is selected from the group consisting of a user-visible sub-statement instruction, a loop entry instruction, a loop body instruction, dead code instruction, and a data-speculative load instruction.

8. (original) The system of claim 6, wherein the unique graphical indicator is selected from the group consisting of text color, background color, a line, an arc, a box and a tag.

9. (original) The system of claim 6, wherein the displaying means further comprises:

means for indicating if the at least one instruction characteristic is a loop-carried dependency.

10. (original) The system of claim 6, wherein the displaying means further comprises: means for indicating if the at least one instruction characteristic is a data-speculative load instruction with at least one possible conflicting store.

11. (original) A computer readable medium for a graphic representation of code characteristics, comprising:

logic for acquiring a block of code in a program;

logic for analyzing the block of code for at least one instruction characteristic;

logic for generating a unique graphical indicator for the at least one instruction characteristic; and

logic for displaying the unique graphical indicator with the block of code to indicate that the at least one instruction characteristic is present in the block of code.

12. (original) The computer readable medium of claim 11, wherein the at least one code characteristic is selected from the group consisting of a user-visible sub-statement instruction, a loop entry instruction, a loop body instruction, dead code instruction, and a data-speculative load instruction.

13. (original) The computer readable medium of claim 11, wherein the unique graphical indicator is selected from the group consisting of text color, background color, a line, an arc, a box and a tag.

14. (original) The computer readable medium of claim 11, wherein the displaying logic further comprises:

logic for indicating if the at least one instruction characteristic is a loop-carried dependency.

15. (original) The computer readable medium of claim 11, wherein the displaying logic further comprises:

logic for indicating if the at least one instruction characteristic is a data-speculative load instruction with at least one possible conflicting store.

16. (currently amended) A system for providing a graphic representation of code characteristics tangibly embodied on a computer readable medium, comprising:

a debug tool that indicates instruction characteristics in a program, wherein the debug tool further comprises:

logic for acquiring a block of code in the program;

logic for analyzing the block of code for the at least one instruction characteristic;

logic for generating a unique graphical indicator for the at least one instruction characteristic; and

logic for displaying the unique graphical indicator with the block of code to indicate that the at least one instruction characteristic is present in the block of code.

17. (original) The system of claim 16, wherein the at least one code characteristic is selected from the group consisting of a user-visible sub-statement instruction, a loop entry instruction, a loop body instruction, dead code instruction, and a data-speculative load instruction.

18. (original) The system of claim 16, wherein the unique graphical indicator is selected from the group consisting of text color, background color, a line, an arc, a box and a tag.

19. (original) The system of claim 16, wherein the displaying logic further comprises:

logic for indicating if the at least one instruction characteristic is a loop-carried dependency.

20. (original) The system of claim 16, wherein the displaying logic further comprises: logic for indicating if the at least one instruction characteristic is a data-speculative load instruction with at least one possible conflicting store.